



Fall 2014 SEI Research Review Verifying Evolving Software

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Arie Gurfinkel
October 28, 2014



Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 28 OCT 2014		2. REPORT TYPE N/A		3. DATES COVERED	
4. TITLE AND SUBTITLE Verifying Evolving Software				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Gurfinkel /Arie				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited.					
13. SUPPLEMENTARY NOTES The original document contains color images.					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 13	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use.

Requests for permission should be directed to the Software Engineering Institute at

permission@sei.cmu.edu.

DM-0001792

Team: Verifying Evolving Software

SEI team members

- Dr. Arie Gurfinkel
- Dr. Sagar Chaki

Collaborators

- Dr. Anton Belov (Synopsys)
- Dr. Nikolaj Bjorner (Microsoft Research)
- Grigory Fedyukovich (Univ. of Lugano)
- Dr. Pierre-Loic Garoche (Onera)
- Dr. Alexander Ivrii (IBM)
- Dr. Temesghen Kahsai (NASA Ames)
- Prof. Natasha Sharygina (University of Lugano)
- Prof. Ofer Strichman (Technion)



Overview

Problem: Scalable verification of evolving software

- reduce re-verification effort
- close *semantic gap* between compiler and verifier
- enable safe use of compiler optimizations in safety-critical code

Related Work: Current solutions are limited by

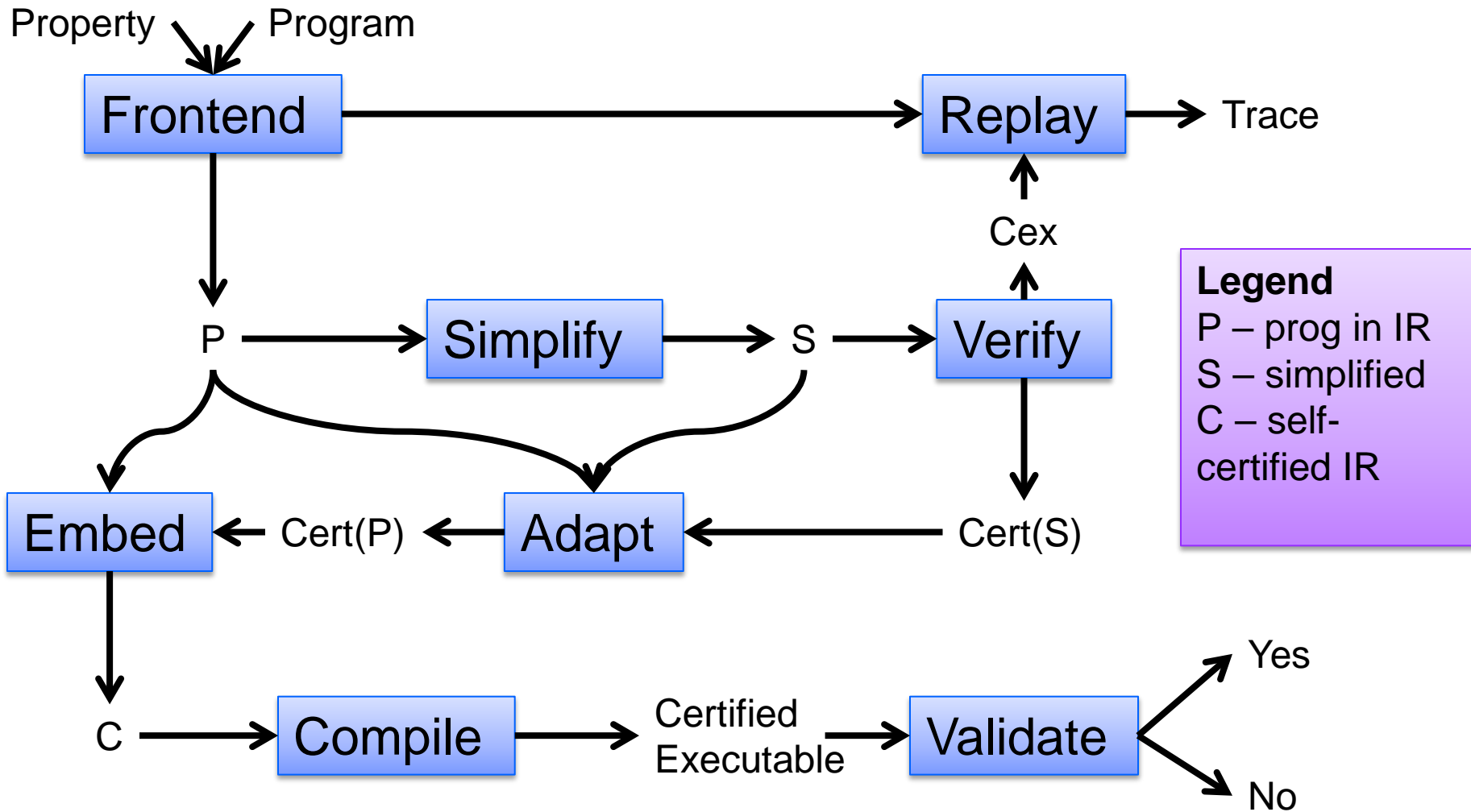
- effectiveness (syntactic slicing, regression verification)
- high-maintenance cost (translation validation)
- narrow applicability (upgrade checking)

Key Idea: Propagate verification certificates across evolution boundaries

- generate verification certificates using *proof-based* verification techniques
- iteratively guess the mapping between original and evolved program
- propagate certificates and strengthen using *incremental inductive verification*
 - IIV is a new verification technique co-developed by us



Model Problem: Certifying Compiler for C



Research Tasks

Verifying instcomine and simplifycfg optimizations of LLVM

- with Prof. Natasha Sharygina and Grigory Fedyukovich (Univ. of Lugano)

Closing the semantic gap between Compiler and Verifier

- with Dr. Anton Belov (Synopsys) and J. Marques-Silva (UCD)

Minimizing verification certificates

- with Dr. Anton Belov (Synopsys) and Dr. Alexander Ivrii (IBM)

Certifying compiler for Luster

- with Dr. Temesghen Kahsai (NASA Ames) and Dr. PL. Garoche (Onera)

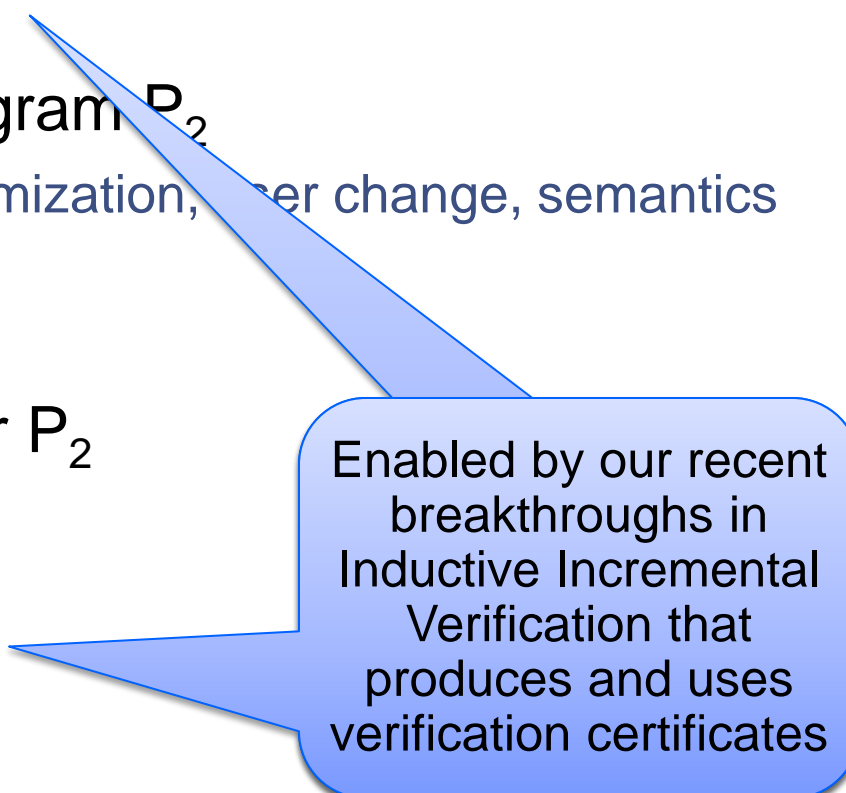
Polyhedral Verification Certificates

- with Dr. Nikolaj Bjorner (Microsoft Research)



Our Approach

1. Compute a verification certificate C_1 for program P_1
2. Evolve program P_1 to a program P_2
 - P_2 is obtained by compiler optimization, user change, semantics change, etc.
3. Adapt C_1 to certificate C_2 for P_2
4. Strengthen C_2 if necessary



Enabled by our recent breakthroughs in Inductive Incremental Verification that produces and uses verification certificates



Research Tasks

Verifying instcomine and simplifycfg optimizations of LLVM

- with Prof. Natasha Sharygina and Grigory Fedyukovich (Univ. of Lugano)

Closing the semantic gap between Compiler and Verifier

- with Dr. Anton Belov (Synopsys) and J. Marques-Silva (UCD)

Minimizing verification certificates

- with Dr. Anton Belov (Synopsys) and Dr. Alexander Ivrii (IBM)

Certifying compiler for Luster

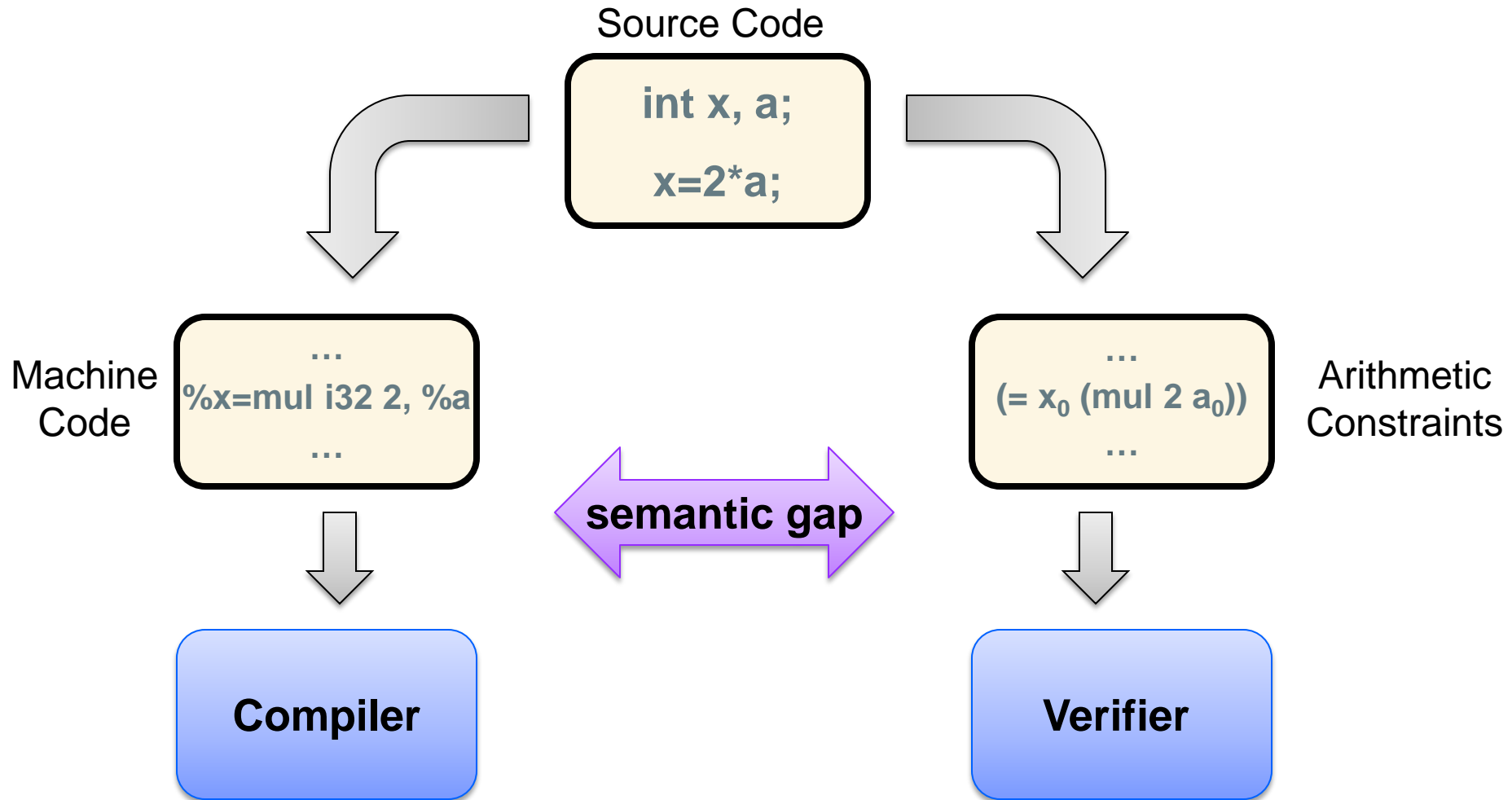
- with Dr. Temesghen Kahsai (NASA Ames) and Dr. PL. Garoche (Onera)

Polyhedral Verification Certificates

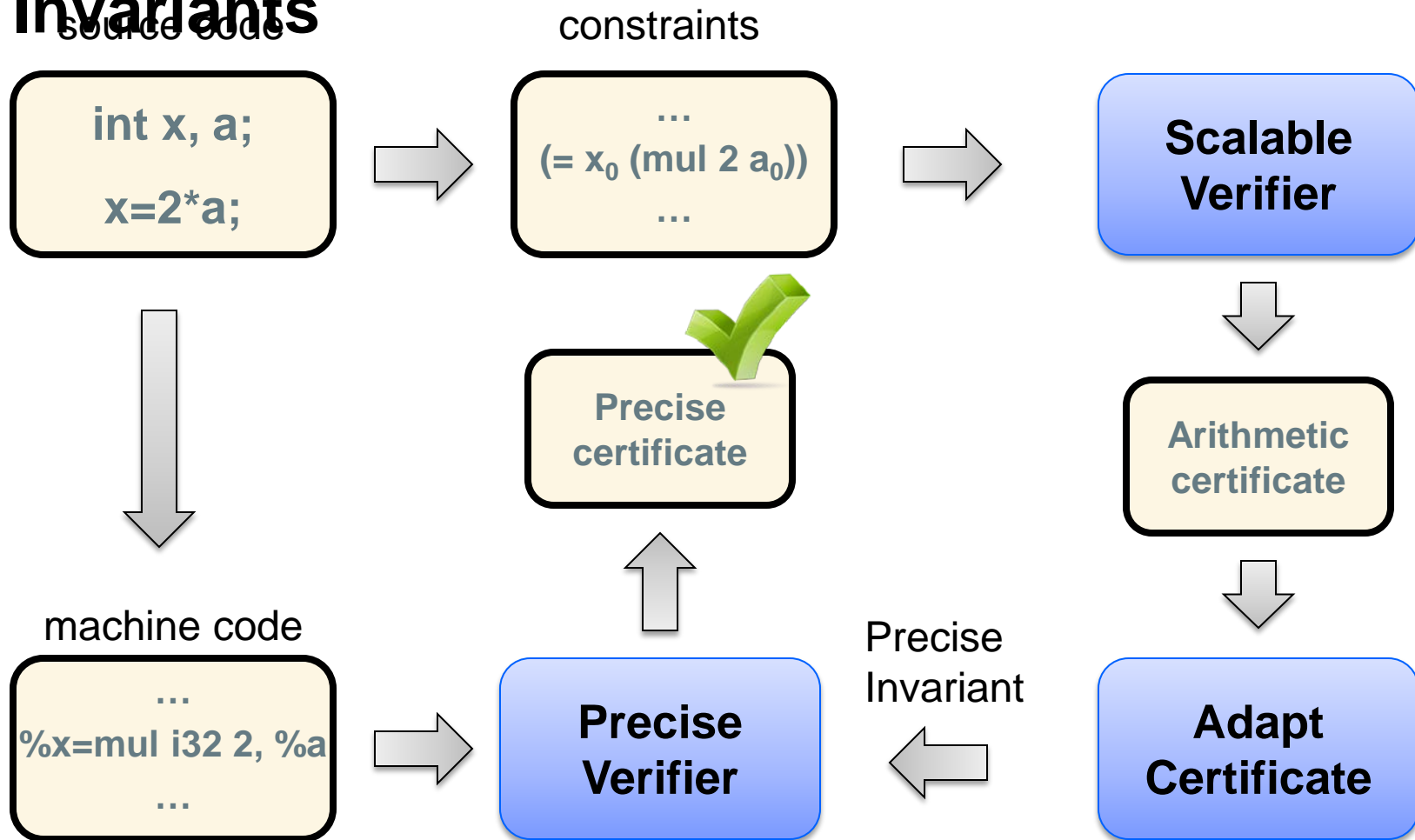
- with Dr. Nikolaj Bjorner (Microsoft Research)



Compiler and Verifier Semantic Gap



MISPER: Synthesizing Safe Bit-Precise Invariants



FrankenBit: Bit-Precise Verification w/ Many Bits

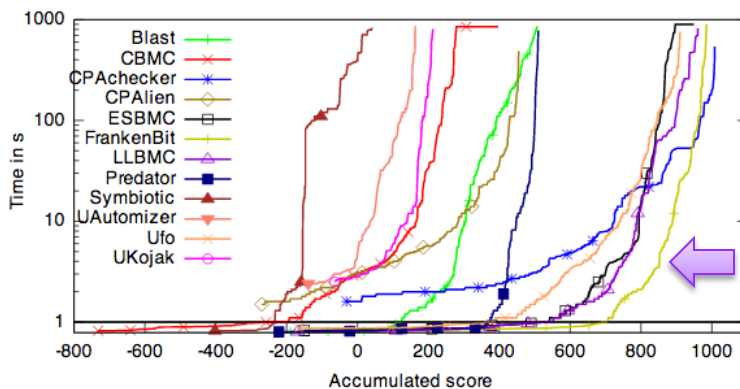
MISPER to synthesize bit-precise invariants

LLBMC to search for counterexamples

Silver and Bronze medals at SV-COMP 2014

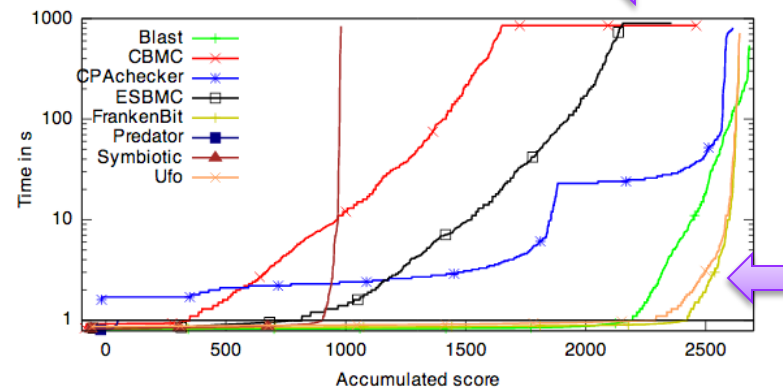
ControlFlow

1. CPAchecker
2. FrankenBit
3. LLBMC



DeviceDrivers64

1. BLAST 2.7.2
2. UFO
3. FrankenBit



<http://sv-comp.sosy-lab.org/2014/results/index.php>



Outcomes

Tools

- FrankenBit – bit-precise verifier for C
- Niagara – validator for LLVM compiler optimizations
- Zuster – verifier for Luster programs

Publications

- Synthesizing Safe Bit-Precise Invariants. TACAS 2014
- FrankenBit: Bit-Precise Verification with Many Bits (Tool paper). TACAS 2014
- Incremental Verification of Compiler Optimizations. NASA FM 2014
- Synthesizing Modular Invariants for Synchronous Code. HCVS 2014
- Small Inductive Safe Invariants. FMCAD 2014
- Property Directed Polyhedral Abstraction. VMCAI 2015
- Automated Discovery of Simulation Between Programs. Submitted to TACAS 2015



Contact Information

Arie Gurfinkel

Sr. Researcher

SSD

Telephone: +1 412-268-5800

Email: arie@sei.cmu.edu

U.S. Mail

Software Engineering Institute

Customer Relations

4500 Fifth Avenue

Pittsburgh, PA 15213-2612

USA

Web

www.sei.cmu.edu/staff/arie

www.sei.cmu.edu/contact.cfm

Customer Relations

Email: info@sei.cmu.edu

Telephone: +1 412-268-5800

SEI Phone: +1 412-268-5800

SEI Fax: +1 412-268-6257

